

## **AMENDMENTS TO THE CLAIMS**

Please amend Claims 1-17 as shown below. This listing of claims will replace all prior versions, and listings, of claims in the international application.

1. *(currently amended)* A method of configuring parameters of an M2M (machine-to-machine) module, the method comprising establishing a connection between the M2M module and a server, ~~characterized by~~ the method further comprising:
  - (304) downloading, to the M2M module, an application having an interface for configuring the M2M module, the application being configured to run on a Java virtual machine (JVM);
  - (306, 308) communicating with the server by the application for receiving configuration parameters; and
  - (310) setting the parameters of the M2M module by the application based on the received configuration parameters.
2. *(currently amended)* The method of claim 1, ~~characterized in that~~ wherein the connection between the M2M module and the server is established through a M2M gateway or over a TCP/IP connection.
3. *(currently amended)* The method of claim 1, ~~characterized in that~~ wherein the communicating with the server comprises: requesting (306) configuration parameters from the server and receiving (308), in the application, the configuration parameters from the server.
4. *(currently amended)* The method of claim 1, ~~characterized by~~ wherein the application comprises being a Java 2 Micro Edition (J2ME<sup>TM</sup>) application.
5. *(currently amended)* The method of claim 3, ~~characterized by~~ wherein the application comprises being a Java MIDlet or a Java IMlet.
6. *(currently amended)* The method of claim 1, ~~characterized by~~ wherein the application being is downloaded to the M2M module over a cable, over an infrared connection or over-the-air (OTA).

7. (currently amended) The method of claim 1, ~~characterized by~~ wherein the application programming interface ~~being~~ is a Common Object Request Broker Architecture (CORBA) API.

8. (currently amended) The method of claim 6 ~~7~~, ~~characterized by~~ wherein the method comprises communicating with the server by making a method call through the CORBA API.

9. (currently amended) The method of claim 1, ~~characterized by~~ wherein the method comprises using the M2M module (110) for configuring parameters of a remote device based on the configuration parameters received from the server.

10. (currently amended) An M2M (machine-to-machine) module (110), comprising: means for operating a Java virtual machine (122) and means (124) for establishing a connection between the M2M module (110) and a server (118),  
~~characterized by~~ wherein the M2M module (110) ~~being~~ is configured to:

download an application (120) having an application programming interface (API) (124) for configuring the M2M module, the application (120) being configured to run on a Java virtual machine (JVM) (122);

communicate with the server (118) by the application (120) for receiving configuration parameters; and

set the parameters of the M2M module (110) by the application (120) based on the received configuration parameters.

11. (currently amended) The M2M module of claim 10, ~~characterized in that~~ wherein the M2M module (110) is further configured to request, by the application, configuration parameters from the server (118) and receiving, in the application, the configuration parameters from the server (118).

12. (currently amended) The M2M module of claim 10, ~~characterized in that~~ wherein the application (120) is a Java 2 Micro Edition (J2ME<sup>TM</sup>) application.

13. *(currently amended)* The M2M module of claim 10, ~~characterized in that~~  
wherein the application (120) is a Java MIDlet or a Java IMlet.

14. *(currently amended)* The M2M module of claim 10, ~~characterized in that~~  
wherein the M2M module (110) is configured to download the application (120) over  
a cable, over an infrared connection or over-the-air (OTA).

15. *(currently amended)* The M2M module of claim 10, ~~characterized in that~~  
wherein the application-programming interface (124) is a Common Object Request  
Broker Architecture (CORBA) API.

16. *(currently amended)* The M2M module of claim 15, ~~characterized in that~~  
wherein the M2M module (110) is configured to communicate with the server (118)  
by making a method call through the CORBA API (124).

17. *(currently amended)* The M2M module of claim 10, ~~characterized in that~~  
wherein the M2M module (110) is used for configuring parameters of a remote device  
(112) based on the configuration parameters received from the server (118).